ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ

Кафедра
«Криптология и кибербезопасность»

Лабораторная работа №1

по предмету «Технологии контейнеризации»

Выполнил студент группы Б20-505

Сорочан Илья

Москва – 2023

Содержание

1. Настройка хоста

В качестве хоста будет использоваться Manjaro:

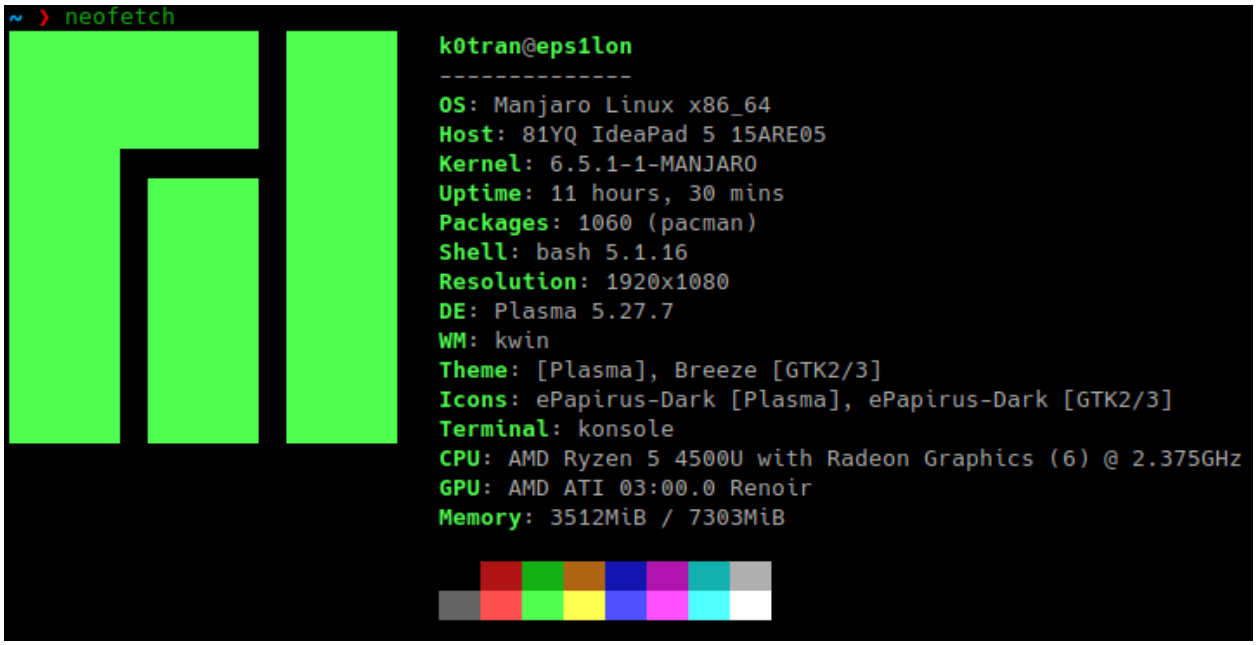

Рисунок 1: Запуск neofetch на хосте

1.1. VirtualBox

Для установки VirtualBox будет использован пакетный менеджер pacman:

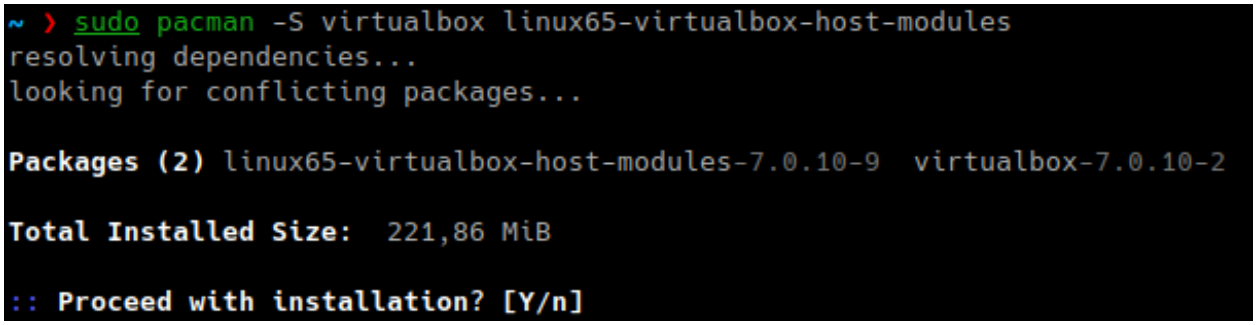

Рисунок 2: Установка VirtualBox

1.2. Vagrant

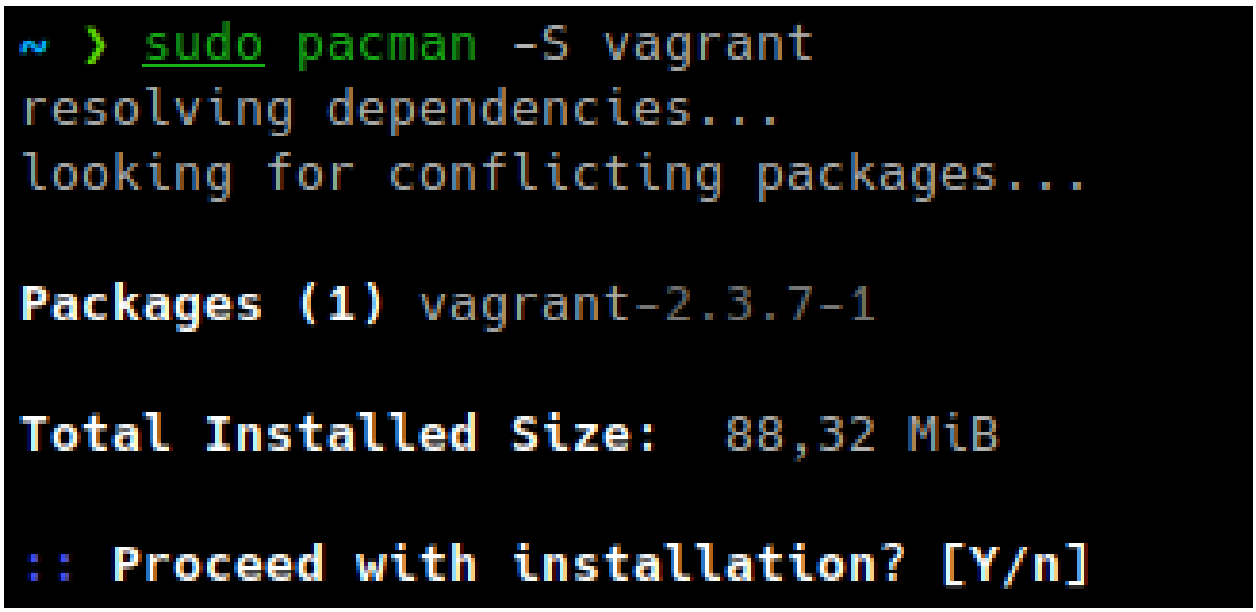

Рисунок 3: Установка Vagrant

Так же следует установить плагин для гостевых дополнений:

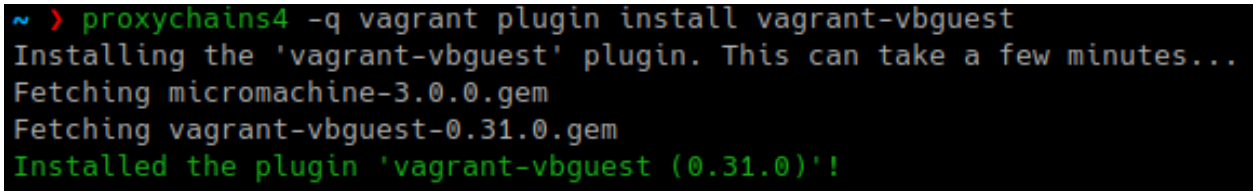

Рисунок 4: Установка гостевых дополнений Vagrant

2. Работа с Vagrant

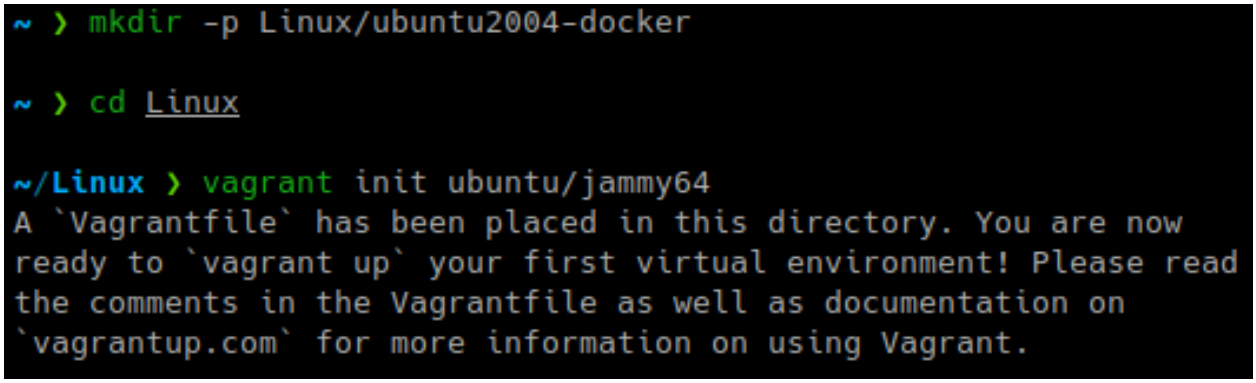

Рисунок 5: Инициализации директории для работы с Vagrant

2.1. Первый запуск

Для того что бы `vagrant up` мог сам скачать виртульную машину первый запуск производится с использованием proxychains:

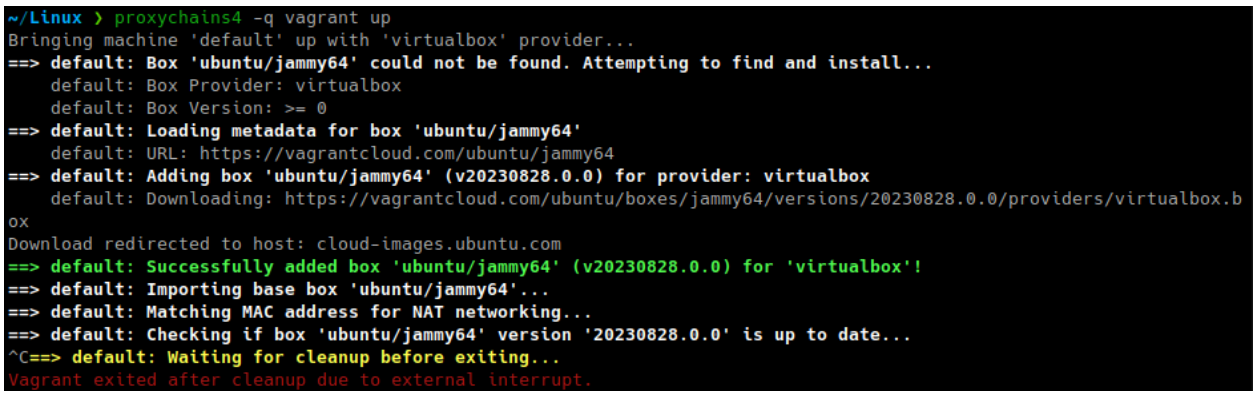

Рисунок 6: Первый запуск vagrant up (прерван)

Можно убедится, что запуск машины успешно прерван:

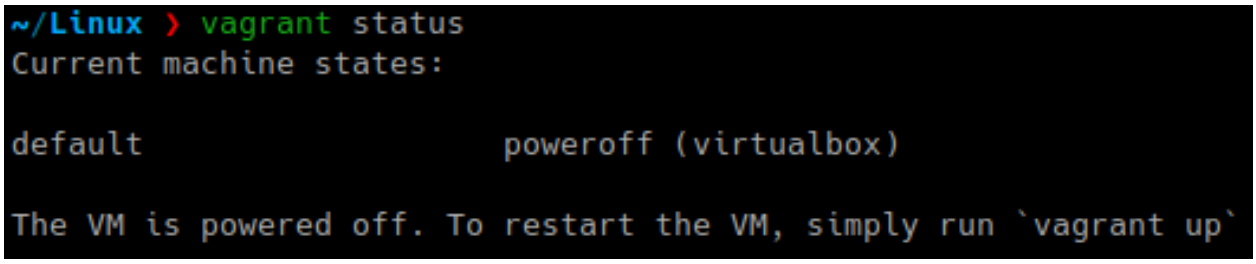

Рисунок 7: vagrant status

Последующие запуски будут производиться без proxychains.

Из-за того, что используется VirtualBox 7.10, в консоль будет выведено следующее предупреждение:

```
$ vagrant up
...
Installing Virtualbox Guest Additions 7.0.10 - guest version is 6.0.0
...
An error occurred during installation of VirtualBox Guest Additions
7.0.10. Some functionality may not work as intended.
In most cases it is OK that the "Window System drivers" installation
failed.
...
```

Остановить и удалить машину можно при помощи vagrant destroy:

```
~/Linux ❯ vagrant destroy
    default: Are you sure you want to destroy the 'default' VM? [y/N] y
==> default: Forcing shutdown of VM...
==> default: Destroying VM and associated drives...
```

Рисунок 8: vagrant destroy

2.2. Настройка Vagrantfile

Для выполнения условий лабораторной следует добавить следующие строчки в Vagrantfile:

```
...
config.vm.hostname = "ubuntu-docker"
...
config.vm.provider "virtualbox" do |vb|
    vb.memory = "2048"
    vb.cpus = 2
    vb.check_guest_additions = false
end
```

Так же будет полезно отключить обновления:

```
...
config.vm.box_check_update = false
...
```

И включить аппаратную виртуализацию:

```
vb.customize ["modifyvm", :id, "--nested-hw-virt", "on"]
```

Создание снимка VM:

```
~/Linux ❯ vagrant snapshot save initial
==> default: Snapshotting the machine as 'initial'...
==> default: Snapshot saved! You can restore the snapshot at any time by
==> default: using `vagrant snapshot restore`. You can delete it using
==> default: `vagrant snapshot delete`.
```

Рисунок 9: Создание снимка VM

Подключение к виртуальной машине по ssh:

```
~/Linux ❯ vagrant ssh
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-82-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Sep 13 08:23:51 UTC 2023

  System load:  0.14599609375     Processes:               110
  Usage of /:   5.3% of 38.70GB   Users logged in:         0
  Memory usage: 13%               IPv4 address for enp0s3: 10.0.2.15
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

11 updates can be applied immediately.
11 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


*** System restart required ***
vagrant@ubuntu-docker:~$ exit
logout
```

Рисунок 10: Подключение по ssh

2.3. Установка Docker Engine

Необходимо добавить репозиторий докера:

```
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg
--dearmor -o /etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

```
# Add the repository to Apt sources:
echo \
   "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/
keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
  "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

Установка докера (пакетов):

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-
buildx-plugin docker-compose-plugin
```

Проверка работоспособности:

```
vagrant@ubuntu-docker:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:4f53e2564790c8e7856ec08e384732aa38dc43c52f02952483e3f003afbf23db
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Рисунок 11: Docker hello-world

2.4. Создание образа ubuntudocker.box

Упаковка и загрузка в локальный репозиторий:

```
~/Linux ❯ vagrant package --output ubuntudocker
==> default: Clearing any previously set forwarded ports...
==> default: Exporting VM...
==> default: Compressing package to: /home/k0tran/Linux/ubuntudocker

~/Linux ❯ vagrant box add --name ubuntudocker ubuntudocker
==> box: Box file was not detected as metadata. Adding it directly...
==> box: Adding box 'ubuntudocker' (v0) for provider:
    box: Unpacking necessary files from: file:///home/k0tran/Linux/ubuntudocker
==> box: Successfully added box 'ubuntudocker' (v0) for 'virtualbox'!
```

Рисунок 12: vagrant package и vagrant box

2.5. Работа со снимками

Создание новой машины:

```
~/Linux ❯ cd ubuntu2004-docker

~/Linux/ubuntu2004-docker ❯ vagrant init ubuntudocker
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.

~/Linux/ubuntu2004-docker ❯ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'ubuntudocker'...
==> default: Matching MAC address for NAT networking...
==> default: Setting the name of the VM: ubuntu2004-docker_default_1694594505159_53619
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
==> default: Forwarding ports...
    default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2222
    default: SSH username: vagrant
    default: SSH auth method: private key
==> default: Machine booted and ready!
[default] GuestAdditions seems to be installed (7.0.10) correctly, but not running.
==> default: Checking for guest additions in VM...
==> default: Mounting shared folders...
    default: /vagrant => /home/k0tran/Linux/ubuntu2004-docker
```

Рисунок 13: Создание новой машины

Создание снимка виртуальной машины:

```
~/Linux/ubuntu2004-docker ❯ vagrant snapshot save default
==> default: Snapshotting the machine as 'default'...
==> default: Snapshot saved! You can restore the snapshot at any time by
==> default: using `vagrant snapshot restore`. You can delete it using
==> default: `vagrant snapshot delete`.
```

Рисунок 14: Создание снимка default

В качестве примера изменения поставим zig:

```
vagrant@ubuntu-docker:~$ sudo snap install zig --classic --beta
zig (beta) 0.11.0 from Jay Petacat (jayschwa) installed
vagrant@ubuntu-docker:~$ zig version
0.11.0
```

Рисунок 15: Создание снимка default

Восстанавливаем снимок default:

```
~/Linux/ubuntu2004-docker ❯ vagrant snapshot restore default
==> default: Forcing shutdown of VM...
==> default: Restoring the snapshot 'default'...
==> default: Resuming suspended VM...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2222
    default: SSH username: vagrant
    default: SSH auth method: private key
==> default: Machine booted and ready!
==> default: Machine already provisioned. Run `vagrant provision` or use the `--provision`
==> default: flag to force provisioning. Provisioners marked to run always will still run.
```

Рисунок 16: Восстановление снимка

Проверяем, что zig не установлен в системе:

```
vagrant@ubuntu-docker:~$ zig
Command 'zig' not found, did you mean:
  command 'kig' from snap kig (23.04.3)
  command 'zim' from deb zim (0.74.3-1)
  command 'zip' from deb zip (3.0-12build2)
  command 'zkg' from deb zkg (2.12.0-1)
  command 'tig' from deb tig (2.5.1-1)
  command 'zic' from deb libc-bin (2.35-0ubuntu3.3)
  command 'dig' from deb bind9-dnsutils (1:9.18.12-0ubuntu0.22.04.2)
  command 'kig' from deb kig (4:21.12.3-0ubuntu1)
  command 'pig' from deb bsdgames (2.17-29)
  command 'wig' from deb wig (0.6-2)
  command 'rig' from deb rig (1.11-1.1)
See 'snap info <snapname>' for additional versions.
```

Рисунок 17: Проверка успешного восстановления